A general and algorithmic method for computing the generating function of permutation classes and for their random generation

Mathilde Bouvel (LaBRI)

avec Frédérique Bassino (LIPN), Adeline Pierrot (LIAFA), Carine Pivoteau (LIGM), Dominique Rossin (LIX)

Séminaire Graphes et structures discrètes, Lyon

Guideline for the talk

Data:

- B a finite set of permutations (the excluded patterns),
- $\mathcal{C} = Av(B)$ the class of permutations that avoid every pattern of B.

Problem:

Describe an algorithm to obtain automatically from B

- some enumerative results on $\mathcal{C}$, in terms of generating function $C(z) = \sum |Av_n(B)| z^n$,
- a random sampler of permutations in $\mathcal{C}$, that is uniform on $Av_n(B)$ for each n.

Result:

Such an algorithm ... that works when $\mathcal{C}$ contains a finite number of simple permutations. Additional algorithms for:

- testing if $\mathcal{C}$ contains a finite number of simple permutations
- computing from B the finite set of simple permutations of $\mathcal{C}$

Outline

Outline

Representation of permutations

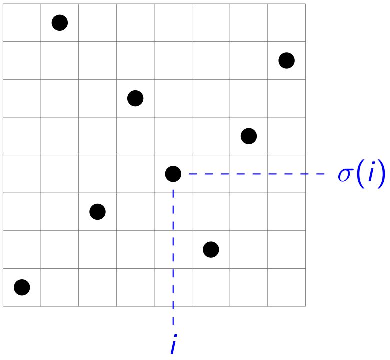
Permutation: Bijection from $[1..n]$ to itself. Set $\mathfrak{S}_n$.

- Linear representation:
  $\sigma = 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7$

- Two lines representation:
  $$\sigma = \begin{pmatrix} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ 1\ 8\ 3\ 6\ 4\ 2\ 5\ 7 \end{pmatrix}$$

- Representation as a product of cycles:
  $\sigma = (1)\ (2\ 8\ 7\ 5\ 4\ 6)\ (3)$

- Graphical representation:



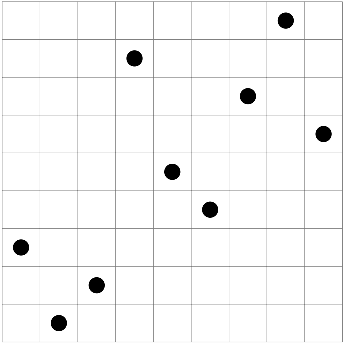
Patterns in permutations

Pattern (order) relation $\preccurlyeq$:
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is order isomorphic ($\equiv$) to π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.



Patterns in permutations

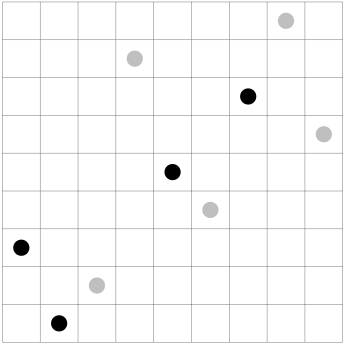
Pattern (order) relation $\preccurlyeq$:

$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\, 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is order isomorphic ($\equiv$) to π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.

Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.

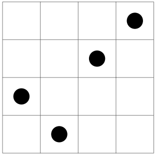


Patterns in permutations

Pattern (order) relation $\preccurlyeq$:
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists\ 1 \leq i_1 < \ldots < i_k \leq n$ such that $\sigma_{i_1} \ldots \sigma_{i_k}$ is order isomorphic ($\equiv$) to π.

Notation: $\pi \preccurlyeq \sigma$.

Equivalently:
The normalization of $\sigma_{i_1} \ldots \sigma_{i_k}$ on $[1..k]$ yields π.



Example: $2\,1\,3\,4 \preccurlyeq \mathbf{3}\,\mathbf{1}\,2\,8\,\mathbf{5}\,4\,\mathbf{7}\,9\,6$
since $3\,1\,5\,7 \equiv 2\,1\,3\,4$.

Permutation classes

Permutation class : set of permutations downward-closed for $\preccurlyeq$.

$Av(B)$: the class of permutations that avoid every pattern of B.
If B is an antichain then B is the basis of $Av(B)$.

Conversely : Every class $\mathcal{C}$ can be characterized by its basis:

$$\mathcal{C} = Av(B) \text{ for } B = \{\sigma \notin \mathcal{C} : \forall \pi \preccurlyeq \sigma \text{ such that } \pi \neq \sigma, \pi \in \mathcal{C}\}$$

A class has a unique basis.
A basis can be either finite or infinite.

Origin : [Knuth 73] with stack-sortable permutations $= Av(231)$

Enumeration[Stanley & Wilf 92][Marcus & Tardos 04] : $|\mathcal{C} \cap \mathfrak{S}_n| \leq c^n$

Problematics

- **Combinatorics**: study of classes defined by their basis.

$\hookrightarrow$ Enumeration.

$\hookrightarrow$ Exhaustive generation.

- **Algorithmics**: problematics from text algorithmics.

$\hookrightarrow$ Pattern matching, longest common pattern.

$\hookrightarrow$ Linked with testing the membership of σ to a class.

- **Combinatorics (and algorithms)**: study families of classes.

$\hookrightarrow$ A class is not always described by its basis.

$\hookrightarrow$ Obtain general results on the structure of a class...

$\hookrightarrow$ ...and do it automatically (with algorithms).

Outline

Substitution decomposition: main ideas

Analogous to the decomposition of integers as products of primes.

- [Möhring & Radermacher 84]: general framework.
- Specialization: Modular decomposition of graphs.

Relies on:

- a principle for building objects (permutations, graphs) from smaller objects: the substitution.
- some "basic objects" for this construction: **simple** permutations, **prime** graphs.

Required properties:

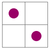
- every object can be decomposed using only "basic objects".
- this decomposition is unique.

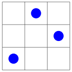
Substitution for permutations

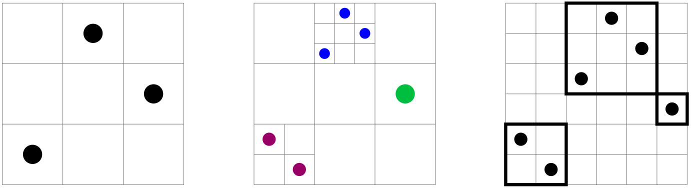
Substitution or inflation : $\sigma = \pi[\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(k)}]$.

Example : Here, $\pi = 1\,3\,2$, and

$$\begin{cases} \alpha^{(1)} = 2\,1 = \\ \alpha^{(2)} = 1\,3\,2 = \\ \alpha^{(3)} = 1 = \end{cases}$$



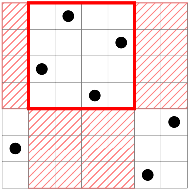




Hence $\sigma = 1\,3\,2[2\,1, 1\,3\,2, 1] = 2\,1\,4\,6\,5\,3$.

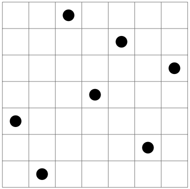
Simple permutations

Interval (or block) = set of elements of σ whose positions **and** values form intervals of integers
Example: $5\,7\,4\,6$ is an interval of $2\,5\,7\,4\,6\,1\,3$



Simple permutation = permutation that has no interval, except the trivial intervals: $1, 2, \ldots, n$ and σ
Example: $3\,1\,7\,4\,6\,2\,5$ is simple.

The smallest simple: $1\,2, 2\,1, 2\,4\,1\,3, 3\,1\,4\,2$



Substitution decomposition of permutations

Theorem: Every σ ($\neq 1$) is uniquely decomposed as

- $12[\alpha^{(1)}, \alpha^{(2)}]$, where $\alpha^{(1)}$ is $\oplus$-indecomposable
- $21[\alpha^{(1)}, \alpha^{(2)}]$, where $\alpha^{(1)}$ is $\ominus$-indecomposable
- $\pi[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where π is simple of size $k \geq 4$

Remarks:

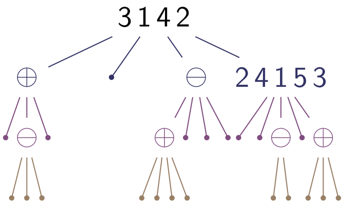
- $\oplus$-indecomposable : that cannot be written as $12[\alpha^{(1)}, \alpha^{(2)}]$
- Result stated as in [Albert & Atkinson 05]
- Can be rephrased changing the first two items into:
  - $12 \dots k[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\oplus$-indecomposable
  - $k \dots 21[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are $\ominus$-indecomposable

Decomposing recursively inside the $\alpha^{(i)}$ $\Rightarrow$ **decomposition tree**

Decomposition tree: witness of this decomposition

Example: Decomposition tree of $\sigma =$

10 13 12 11 14 1 18 19 20 21 17 16 15 4 8 3 2 9 5 6 7



Notations and properties:

- $\oplus = 12\dots k$ and $\ominus = k\dots 21$ = linear nodes.
- π simple of size ≥ 4 = prime node.
- No edge $\oplus - \oplus$ nor $\ominus - \ominus$.
- Ordered trees.

Expansion of $\oplus[T_1, T_2, T_3, \dots]$ into $\oplus[T_1, \oplus[T_2, T_3, \dots]]$ and recursively, for the version of the trees of [AA05]

$$\sigma = 3\,1\,4\,2[\oplus[1, \ominus[1, 1, 1], 1], 1, \ominus[\oplus[1, 1, 1, 1], 1, 1, 1], 2\,4\,1\,5\,3[1, 1, \ominus[1, 1], 1, \oplus[1, 1, 1]]]$$

Bijection between permutations and their decomposition trees.

Computation and examples of application

Computation: in linear time. [Uno & Yagiura 00] [Bui Xuan, Habib & Paul 05] [Bergeron, Chauve, Montgolfier & Raffinot 08]

In algorithms:

- Pattern matching [Bose, Buss & Lubiw 98] [Ibarra 97]
- Algorithms for bio-informatics [Bérard, Bergeron, Chauve & Paul 07] [Bérard, Chateau, Chauve, Paul & Tannier 08]

In combinatorics:

- Simple permutations [Albert, Atkinson & Klazar 03]
- Classes closed by substitution product [Atkinson & Stitt 02] [Brignall 07] [Atkinson, Ruškuc & Smith 09]
- Exhibit the structure of classes [Albert & Atkinson 05] [Brignall, Huczynska & Vatter 08a,08b] [Brignall, Ruškuc & Vatter 08]

Outline

Combinatorial classes and generating functions

Notations:

- $\mathcal{C} = \cup_{n\geq 0}\mathcal{C}_n$ with finite number $c_n = |\mathcal{C}_n|$ of objects of size n
- Generating function $C(z) = \sum c_n z^n$

Recursive description with constructors $\Rightarrow$ Equation on the g.f.:

Constructor	Notation	$C(z)$
Atom	$\mathcal{Z}$	z
Disjoint Union	$\mathcal{A} + \mathcal{B}$	$A(z) + B(z)$
Cartesian Product	$\mathcal{A} \times \mathcal{B}$	$A(z)B(z)$
Sequence	$\textsc{Seq}(\mathcal{A})$	$\dfrac{1}{1 - A(z)}$
Restricted Seq.	$\textsc{Seq}_{=k}(\mathcal{A})$	$A(z)^k$

[Flajolet & Sedgewick 09]

Combinatorial classes and random samplers

Uniform sampling: objects of size n have the same probability

Two methods based on the recursive description of objects:

- Recursive method [Flajolet, Zimmerman & Van Cutsem 94]: size n chosen in advance. Requires to know the c_k for $k \leq n$.
- **Boltzmann method** [Duchon, Flajolet, Louchard & Schaeffer 04]: size n not fixed. Needs the evaluation of $C(z)$ at one point x.

$\mathcal{Z}$	return an atom
$\mathcal{A}+\mathcal{B}$	call $\Gamma A(x)$ with proba. $\frac{A(x)}{A(x)+B(x)}$, else $\Gamma B(x)$
$\mathcal{A}\times\mathcal{B}$	call $\Gamma A(x)$ and $\Gamma B(x)$
$\textsc{Seq}(\mathcal{A})$	choose k according to a geometric law of parameter $A(x)$ and call $\Gamma A(x)$ k times
$\textsc{Seq}_{=k}(\mathcal{A})$	call the sampler $\Gamma A(x)$ k times

Example: binary trees

$\mathcal{B} = \cup_{n\geq 1}\mathcal{B}_n$
where $\mathcal{B}_n$ denotes the set of binary trees with n leaves.

Recursive description (also called specification): $\mathcal{B} = \bullet \quad + \quad \mathcal{B} \quad \mathcal{B}$

Equation for the g.f.: $B(z) = z + B(z)^2$, hence $B(z) = \frac{1-\sqrt{1-4z}}{2}$.

Boltzmann random sampler $\Gamma\mathcal{B}(x)$ for $\mathcal{B}$:

- Data: x, $B(x)$
- Result: a random binary tree
- Procedure:
  - Choose r uniformly at random on $[0,1]$
  - If $r < \frac{x}{B(x)}$ then return $\bullet$
  - Else return $\Gamma\mathcal{B}(x)\,\Gamma\mathcal{B}(x)$

Specifications for permutation classes

For all permutations, with $\mathcal{S}$ the set of all simple permutations:

$$\begin{cases} \mathfrak{S} = \bullet + \overset{\oplus}{\mathfrak{S}^+ \;\; \mathfrak{S}} + \overset{\ominus}{\mathfrak{S}^- \;\; \mathfrak{S}} + \sum_{\pi \in \mathcal{S}} \overset{\pi}{\mathfrak{S} \;\; \mathfrak{S} \;\; \cdots \;\; \mathfrak{S}} \\ \mathfrak{S}^+ = \bullet + \overset{\ominus}{\mathfrak{S}^- \;\; \mathfrak{S}} + \sum_{\pi \in \mathcal{S}} \overset{\pi}{\mathfrak{S} \;\; \mathfrak{S} \;\; \cdots \;\; \mathfrak{S}} \\ \mathfrak{S}^- = \bullet + \overset{\oplus}{\mathfrak{S}^+ \;\; \mathfrak{S}} + \sum_{\pi \in \mathcal{S}} \overset{\pi}{\mathfrak{S} \;\; \mathfrak{S} \;\; \cdots \;\; \mathfrak{S}} \end{cases}$$

$\Rightarrow$The generating functions of $\mathfrak{S}$ and $\mathcal{S}$ are related [Albert, Atkinson & Klazar 03].

This can be adapted to (substitution-closed and arbitrary) permutation classes [Albert & Atkinson 05].

The simpler case of substitution-closed classes

A permutation class $\mathcal{C}$ is **substitution-closed** when $\pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}] \in \mathcal{C}$ for all $\pi, \alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)} \in \mathcal{C}$.

Hence, with $\mathcal{S}_\mathcal{C} = \mathcal{C} \cap \mathcal{S}$ the set of simple permutations in $\mathcal{C}$:

$$\begin{cases} \mathcal{C} = \bullet + \overset{\oplus}{\mathcal{C}^+ \quad \mathcal{C}} + \overset{\ominus}{\mathcal{C}^- \quad \mathcal{C}} + \sum_{\pi \in \mathcal{S}_\mathcal{C}} \overset{\pi}{\mathcal{C} \quad \mathcal{C} \quad \dots \quad \mathcal{C}} \\ \dots \end{cases}$$

When $\mathcal{S}_\mathcal{C}$ is finite, this is a simple family of trees in the sense of [Flajolet & Sedgewick 09].

$\Rightarrow$Enumerative results and random samplers can be obtained by efficient algorithms.

For general permutation classes

For non substitution-closed classes, we have only a strict inclusion:

$$\mathcal{C} \subsetneq \bullet + \overset{\oplus}{\mathcal{C}^+ \quad \mathcal{C}} + \overset{\ominus}{\mathcal{C}^- \quad \mathcal{C}} + \sum_{\pi \in \mathcal{S}_{\mathcal{C}}} \overset{\pi}{\mathcal{C} \quad \mathcal{C} \quad \cdots \quad \mathcal{C}}$$

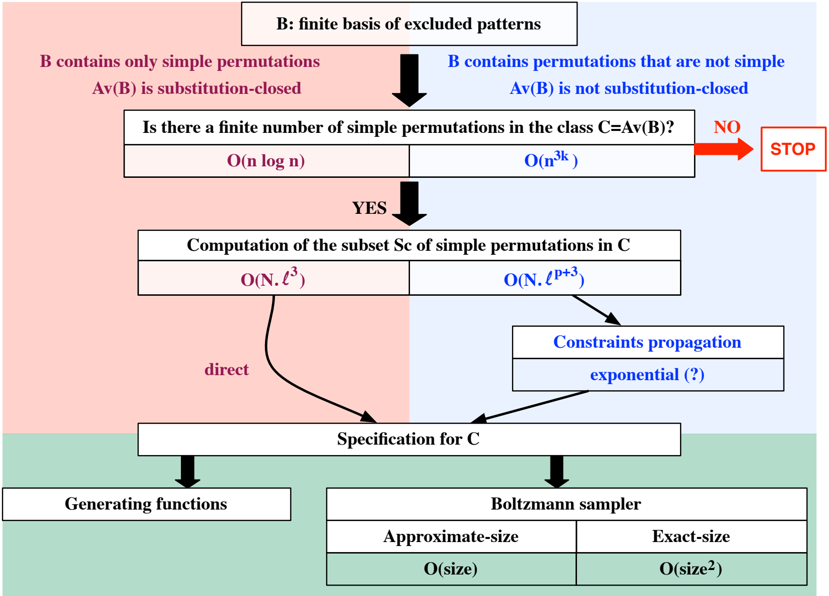
Example: $\ominus[12, 1] \notin Av(231)$ whereas $12, 1 \in Av(231)$.

The system describing $\mathcal{C}$ has to be refined with new equations for these constraints. The system can be computed by an algorithm.

$\Rightarrow$Enumerative results and random samplers can be obtained algorithmically, but this is less efficient.

Outline

Summary of the overall procedure



From the simple permutations to the specification for $\mathcal{C}$

From the set $\mathcal{S}_\mathcal{C}$ of simple permutations in $\mathcal{C}$, the specification for the substitution closure $\hat{\mathcal{C}}$ of $\mathcal{C}$ is obtained immediately:

$$\begin{cases} \hat{\mathcal{C}} = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}] + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}] \\ \hat{\mathcal{C}}^+ = \bullet + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}] \\ \hat{\mathcal{C}}^- = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}] \end{cases}$$

- If $\mathcal{C}$ is substitution-closed, $\mathcal{C} = \hat{\mathcal{C}}$ and we are done.
- Otherwise, $\mathcal{C} = \hat{\mathcal{C}}\langle B^\star\rangle$ and propagate the constraints from $B^\star = \{\beta \in B \ : \ \beta \text{ is not simple } \}$ into the subtrees.

From the simple permutations to the specification for $\mathcal{C}$

From the set $\mathcal{S}_{\mathcal{C}}$ of simple permutations in $\mathcal{C}$, the specification for the substitution closure $\hat{\mathcal{C}}$ of $\mathcal{C}$ is obtained immediately:

$$\begin{cases}
\hat{\mathcal{C}}\langle B^\star\rangle = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}\langle B^\star\rangle] + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle B^\star\rangle] + \sum_{\pi\in\mathcal{S}_{\mathcal{C}}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}\langle B^\star\rangle] \\
\hat{\mathcal{C}}^+\langle B^?\rangle = \bullet + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle B^?\rangle] + \sum_{\pi\in\mathcal{S}_{\mathcal{C}}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}\langle B^?\rangle] \\
\hat{\mathcal{C}}^-\langle B^?\rangle = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}\langle B^?\rangle] + \sum_{\pi\in\mathcal{S}_{\mathcal{C}}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \dots, \hat{\mathcal{C}}\langle B^?\rangle]
\end{cases}$$

- If $\mathcal{C}$ is substitution-closed, $\mathcal{C} = \hat{\mathcal{C}}$ and we are done.
- Otherwise, $\mathcal{C} = \hat{\mathcal{C}}\langle B^\star\rangle$ and propagate the constraints from $B^\star = \{\beta \in B \ : \ \beta \text{ is not simple } \}$ into the subtrees.

From the simple permutations to the specification for $\mathcal{C}$

From the set $\mathcal{S}_\mathcal{C}$ of simple permutations in $\mathcal{C}$, the specification for the substitution closure $\hat{\mathcal{C}}$ of $\mathcal{C}$ is obtained immediately:

$$\begin{cases} \hat{\mathcal{C}}\langle B^?\rangle = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}\langle B^?\rangle] + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle B^?\rangle] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \ldots, \hat{\mathcal{C}}\langle B^?\rangle] \\ \hat{\mathcal{C}}^+\langle B^?\rangle = \bullet + \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle B^?\rangle] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \ldots, \hat{\mathcal{C}}\langle B^?\rangle] \\ \hat{\mathcal{C}}^-\langle B^?\rangle = \bullet + \oplus[\hat{\mathcal{C}}^+, \hat{\mathcal{C}}\langle B^?\rangle] + \sum_{\pi\in\mathcal{S}_\mathcal{C}} \pi[\hat{\mathcal{C}}, \hat{\mathcal{C}}, \ldots, \hat{\mathcal{C}}\langle B^?\rangle] \end{cases}$$

- If $\mathcal{C}$ is substitution-closed, $\mathcal{C} = \hat{\mathcal{C}}$ and we are done.
- Otherwise, $\mathcal{C} = \hat{\mathcal{C}}\langle B^\star\rangle$ and propagate the constraints from $B^\star = \{\beta \in B \ : \ \beta \text{ is not simple }\}$ into the subtrees.

Pushing constraints into the subtrees

Embeddings of $\beta \in B^{\star}$ into $\pi \in \mathcal{S}_{\mathcal{C}} \cup \{12, 21\}$

- Example:
  for $\ominus[\mathcal{C}^{-}, \mathcal{C}]\langle 3412 \rangle$, there are 3 embeddings of 3412 into 21, $(34, 12) \hookrightarrow (2, 1)$ and the trivial ones $(3412, \emptyset) \hookrightarrow (2, 1)$ and $(\emptyset, 3412) \hookrightarrow (2, 1)$.
- additional restrictions α in $B^{?}$ that are blocks of $\beta \in B^{\star}$
- and do it inductively while new constraints α appear
- this terminates since each $\alpha \preccurlyeq \beta$ for some $\beta \in B^{\star}$

Pushing constraints into the subtrees

Embeddings of $\beta \in B^\star$ into $\pi \in \mathcal{S}_\mathcal{C} \cup \{12, 21\}$

- Example:
  for $\ominus[\mathcal{C}^-, \mathcal{C}]\langle 3412 \rangle$, there are 3 embeddings of 3412 into 21, $(34, 12) \hookrightarrow (2, 1)$ and the trivial ones $(3412, \emptyset) \hookrightarrow (2, 1)$ and $(\emptyset, 3412) \hookrightarrow (2, 1)$.
- additional restrictions α in $B^?$ that are blocks of $\beta \in B^\star$
- and do it inductively while new constraints α appear
- this terminates since each $\alpha \preccurlyeq \beta$ for some $\beta \in B^\star$

Result: A system describing $\mathcal{C}$, that may be ambiguous

An ambiguous system

Example: At the first step for $\ominus[\mathcal{C}^-, \mathcal{C}]\langle 3412\rangle$, we get:

$$\ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 3412\rangle] = \ominus[\hat{\mathcal{C}}^-\langle 3412\rangle, \hat{\mathcal{C}}] \cap \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 3412\rangle] \cap \left(\ominus[\hat{\mathcal{C}}^-\langle 12\rangle, \hat{\mathcal{C}}] \cup \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 12\rangle]\right)$$

$$= \ominus[\hat{\mathcal{C}}^-\langle 12\rangle, \hat{\mathcal{C}}\langle 3412\rangle] \cup \ominus[\hat{\mathcal{C}}^-\langle 3412\rangle, \hat{\mathcal{C}}\langle 12\rangle]$$

An ambiguous system

Example: At the first step for $\ominus[\mathcal{C}^-, \mathcal{C}]\langle 3412\rangle$, we get:

$$\ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 3412\rangle] = \ominus[\hat{\mathcal{C}}^-\langle 3412\rangle, \hat{\mathcal{C}}] \cap \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 3412\rangle] \cap \left(\ominus[\hat{\mathcal{C}}^-\langle 12\rangle, \hat{\mathcal{C}}] \cup \ominus[\hat{\mathcal{C}}^-, \hat{\mathcal{C}}\langle 12\rangle]\right)$$

$$= \ominus[\hat{\mathcal{C}}^-\langle 12\rangle, \hat{\mathcal{C}}\langle 3412\rangle] \cup \ominus[\hat{\mathcal{C}}^-\langle 3412\rangle, \hat{\mathcal{C}}\langle 12\rangle]$$

Rem. 1 The new excluded pattern 12 appears, and this new constraint should be further pushed into the substrees.

Rem. 2 The two terms of the union have a non-empty intersection $\Rightarrow$ Need of disambiguation.

Disambiguation of the system

- Use formulas of the type $A \cup B = A \cap B \;\uplus\; \overline{A} \cap B \;\uplus\; A \cap \overline{B}$
- In complement set, excluded patterns become mandatory patterns: $\mathcal{C}_{\gamma}$ for $\gamma \preccurlyeq \beta \in B^{\star}$
- Propagate also mandatory restrictions

Example: From $\ominus\big(\hat{\mathcal{C}}^{-},\ \hat{\mathcal{C}}\langle 3412\rangle\big) = \ominus\big(\hat{\mathcal{C}}^{-}\langle 12\rangle,\ \hat{\mathcal{C}}\langle 3412\rangle\big) \cup \ominus\big(\hat{\mathcal{C}}^{-}\langle 3412\rangle,\ \hat{\mathcal{C}}\langle 12\rangle\big)$,
we obtain:

$$\ominus\big(\hat{\mathcal{C}}^{-},\ \hat{\mathcal{C}}\langle 3412\rangle\big) = \ominus\big(\hat{\mathcal{C}}^{-}\langle 12\rangle,\ \hat{\mathcal{C}}\langle 12\rangle\big) \uplus \ominus\big(\hat{\mathcal{C}}^{-}_{12}\langle 3412\rangle,\ \hat{\mathcal{C}}\langle 12\rangle\big) \uplus \ominus\big(\hat{\mathcal{C}}^{-}\langle 12\rangle,\ \hat{\mathcal{C}}_{12}\langle 3412\rangle\big).$$

Notice that the terms $\ominus\big(\hat{\mathcal{C}}^{-}\langle 3412\rangle,\ \hat{\mathcal{C}}_{3412}\langle 12\rangle\big)$ and $\ominus\big(\hat{\mathcal{C}}^{-}_{3412}\langle 12\rangle,\ \hat{\mathcal{C}}\langle 3412\rangle\big)$ are empty, and have been deleted.

Disambiguation of the system

Result: An unambiguous system (i.e. a combinatorial specification) describing $\mathcal{C}$, where the left-hand-sides are $\mathcal{C}^{\varepsilon}_{\gamma_1,\ldots,\gamma_p}\langle\alpha_1,\ldots,\alpha_k\rangle$ with $\varepsilon \in \{\ \ ,+,-\}$.

Termination: all α_i and γ_j are patterns of some $\beta \in B^\star$

Theorem: The propagation of the constraints to obtain a specification for $\mathcal{C}$ is algorithmic, but there is an explosion of the number of equations in the system.

Open question: provide bounds on the number of equations of the system produced.

Outline

What next?

About the algorithm:

- Implementation in progress
- Complexity analysis of step 3 (explosion of the system)
- Dependency of the complexity of Boltzmann random samplers w.r.t. the size of the specification

With the algorithm:

- From the specifications, estimate growth rates of classes
- Are random permutations in $\mathcal{C}$ “like” in $\mathfrak{S}$?
- Compare statistics on $\mathcal{C}$ and $\mathfrak{S}$, or on $\mathcal{C}_1$ and $\mathcal{C}_2$

Related questions:

- How general is our algorithm?
- Classes with infinite set of simples, but finitely described?
- Use specification of a class to decide membership efficiently?

Almost 30 000 permutations of size 500 in $Av(2413, 1243, 2341, 531642, 41352)$